Algorithmica (2013) 65:43–59
DOI 10.1007/s00453-011-9572-8

Mapping Simple Polygons: How Robots Benefit from Looking Back

Jérémie Chalopin · Shantanu Das · Yann Disser · Matúš Mihalák · Peter Widmayer

Received: 14 December 2010 / Accepted: 9 September 2011 / Published online: 14 October 2011

Abstract We consider the problem of mapping an initially unknown polygon of size n with a simple robot that moves inside the polygon along straight lines between the vertices. The robot sees distant vertices in counter-clockwise order and is able to recognize the vertex among them which it came from in its last move, i.e. the robot can *look back*. Other than that the robot has no means of distinguishing distant vertices. We assume that an upper bound on n is known to the robot beforehand and show that it can always uniquely reconstruct the visibility graph of the polygon. Additionally, we show that multiple identical and deterministic robots can always solve the weak rendezvous problem in which the robots need to position themselves such that all of them are mutually visible to each other.

Our results are tight in the sense that the strong rendezvous problem, where robots need to gather at a vertex, cannot be solved in general, and, without knowing a bound beforehand, not even n can be determined. In terms of mobile agents exploring a

A preliminary version of this paper appeared in the Proceedings of the 7th International Conference on Algorithms and Complexity (CIAC 2010) [7].

J. Chalopin · S. Das
LIF, CNRS & Aix-Marseille Université, Marseille, France

J. Chalopin
e-mail: jeremie.chalopin@lif.univ-mrs.fr

S. Das
e-mail: shantanu.das@acm.org

Y. Disser · M. Mihalák (✉) · P. Widmayer
Institute of Theoretical Computer Science, ETH Zurich, Zurich, Switzerland
e-mail: matus.mihalak@inf.ethz.ch

Y. Disser
e-mail: ydisser@inf.ethz.ch

P. Widmayer
e-mail: widmayer@inf.ethz.ch

graph, our result implies that they can reconstruct any graph that is the visibility graph of a simple polygon. This is in contrast to the known result that the reconstruction of arbitrary graphs is impossible in general, even if n is known.

Keywords Polygon · Mapping · Visibility graph · Rendezvous · Robot

1 Introduction

We are interested in the study of simple but autonomous robots that are capable of performing tasks like searching, cleaning, guarding or gathering data in unknown environments. One of the most fundamental problems involved in these tasks is to obtain a map of the environment. The complexity of this problem depends on the environment in which a robot operates, and on the robot's sensors. For example, it makes a difference whether there are identifiable landmarks in the environment, or whether a robot has access to powerful sensors like GPS. In some settings, large numbers of simple robots are deployed in an environment and expected to cooperate and solve given tasks. A fundamental problem that arises in many settings with multiple robots is the *rendezvous* problem in which robots either need to gather at the same location (strong rendezvous) or at least need to position themselves such that they are visible to each other (weak rendezvous).

A challenging computational and algorithmic question in this context is the study of simple autonomous robots that have only "weak" sensors and only basic moving capabilities. The problem is to decide what capabilities are sufficient for solving fundamental tasks like those mentioned above. The goal is to characterize the difficulty of such a task by finding *minimal* combinations of capabilities that enable a simple robot (or multiple robots) to perform the task.

The task of mapping an unknown environment has previously been studied both in graphs and in geometric environments where robots move in the plane. In this paper we focus on the latter case, more precisely on the case of robots operating in polygonal environments. For many tasks a geometrically accurate map of the environment is not needed, instead a qualitative description of its shape is sufficient. The *visibility graph* of a polygon is such a qualitative map: Its nodes are the vertices of the polygon and two nodes are joined by an edge if they are visible to each other, i.e. if the line segment connecting them lies entirely within the polygon. Visibility graphs and their characterization have been studied extensively [13]. A general goal is to find minimalistic configurations of sensors and movement capabilities that enable simple robots to reconstruct the visibility graph of a polygon, and, in the case of multiple robots, to solve the rendezvous problem.

The rendezvous problem has been solved for robots that freely move in the plane (without polygon) and measure distances [17]. Similar studies in this setting consider for example robots with limited visibility and memory [2] or robots subject to measurement errors [9]. There have also been several studies concerned with the capabilities of minimalistic robots in polygonal environments. For instance it has been shown that a single pebble alone already enables a simple robot to reconstruct the visibility graph [16] of its environment. Other models for simplistic robots have been

studied (e.g. [6, 12, 19]). The variant considered in this paper originates from [16]. Roughly speaking, we allow a robot to order the vertices it sees in counter-clockwise order and to drive to any vertex it sees (for more details cf. Sect. 2). It was previously shown that without initial knowledge about the size of the polygon, such a robot cannot infer the size of a polygon and thus cannot reconstruct its visibility graph [6]. If the robot is only allowed to move along the boundary of the polygon, it cannot reconstruct the visibility graph, even if the number of vertices n is known beforehand [4]. It is still an open question whether the knowledge of n helps in the general case where the robot is not restricted to move along the boundary.

We extend the basic model by allowing the robot to *look back* after moving, i.e. to know which vertex it came from among those now visible to it. In [4] another sensor was used in combination with the look-back capability, and an algorithm for reconstructing the visibility graph was given for a robot equipped with both sensors, for the case in which the total number of vertices n is not known beforehand (not even an upper bound on n). The look-back capability alone, on the other hand, is not even sufficient for the robot to infer n [6]. In this paper we show that if an upper bound on n is known beforehand, a robot with look-back capability can reconstruct the visibility graph of any simple polygon and thus, in particular, determine its size. Furthermore, we show that the look-back capability is sufficient for solving the weak rendezvous problem in all simple polygons if an upper bound on n is known. This is in contrast to the fact that the strong rendezvous problem cannot always be solved, even when the geometry of the polygon is known.

The above results are for simple polygons and do not generalize to polygons with holes. In the latter case, knowing an upper bound on the number of vertices is not sufficient to solve the polygon reconstruction problem: The vertices of a triangle with slightly smaller inscribed triangular hole cannot be distinguished from the vertices of a square with a large square hole. This means it is not even possible to infer n precisely from an upper bound, and thus, in particular, the visibility graph cannot be reconstructed. Moreover, even if the exact value of n (or even the visibility graph) is known, the weak rendezvous problem cannot be solved in polygons with holes. The question whether visibility graph reconstruction becomes possible when n is known is still open.

The problem of reconstructing the visibility graph of a polygon can be seen as a special instance of the problem of reconstructing a general graph. In a graph the robot is allowed to move along edges and sense the degree of the vertex it is currently located at. A usual assumption is that the edges incident to a vertex are distinctly labeled by a *port-numbering* function. The resulting model is equivalent to the *message-passing network model* of distributed computation [8]. A characterization of the graphs for which the rendezvous problem is solvable has been given [8, 18]. From the results of [18] it is easy to infer that there are graphs which cannot be distinguished by a robot, i.e. it is not always possible to reconstruct a graph. A robot with look-back capability moving in a polygon can imitate an agent that is moving in the corresponding visibility graph, which allows us to borrow concepts from the setting for general graphs. We are able to show that while the graph reconstruction problem and the rendezvous problem are not solvable in the general setting, the reconstruction problem as well as the weak rendezvous problem are solvable for visibility graphs

if a bound on n is known and if the port-numbering gives a way to find the counter-clockwise order of the edges as they appear in the underlying polygon, starting from the edge corresponding to the polygon boundary.

Related Work There have been several papers concerned with the capabilities of minimalistic robots (e.g. [12, 16, 19]). The variant considered in this paper originates from [16]. As mentioned above, the ability to look back was introduced in [4], but no results without additional capabilities were previously shown.

The rendezvous problem for robots [1] gained a lot of interest, mainly in the setting of robots moving in the unbounded plane, where every robot can measure the exact distance to every other robot [2, 9, 17]. In this setting, all robots are required to reach a common location or at least converge towards a common location. The problem has been studied in many different variations, e.g. in the asynchronous [11] or the semi-synchronous model [17], with memory or without [2], under limited visibility [2, 11] or unbounded visibility [9, 17].

The rendezvous problem for robots operating in unlabeled graphs has been extensively studied in the setting of distributed computing (cf. [14] for a survey). These studies are related to the more general question of what can be computed in a fixed network of processors that are indistinguishable from each other. This fundamental question was first raised by Angluin [3] and later studied by Boldi et al. [5] and Yamashita and Kameda [18], who in particular gave a complete characterization of graphs where problems like leader election and map construction are solvable (deterministically). These results carry over to the model of mobile agents/robots moving on a graph, due to the equivalence of the two models and the similarity of the leader election and rendezvous problems.

2 Notation and Basic Properties of Polygons

Polygons and Visibility Graphs In this work we consider simple polygons only and we assume polygons to be in general position, i.e. no three vertices lie on a common line. Let in the following $\mathcal{P}$ be such a simple polygon with n vertices. We denote the set of vertices of $\mathcal{P}$ by $V(\mathcal{P})$, where we drop the argument $\mathcal{P}$ whenever the polygon is evident from the context. The boundary of $\mathcal{P}$ together with an (arbitrary) choice of a starting vertex v_0 induces an order among the vertices and we write $v_0, \ldots, v_{n-1}$ to denote the vertices along the boundary in counter-clockwise order. We define $\text{chain}(v_i, v_j) := (v_i, v_{i+1}, \ldots, v_j)$. Note that throughout this paper all indices of vertices are modulo n, unless otherwise specified.

We say two vertices $u, w \in V$ *see* each other or u *sees* w if and only if the line segment $\overline{uw}$ lies entirely in $\mathcal{P}$—in particular v_i sees v_{i+1} for all i. If v_{i-1} sees v_{i+1}, we say v_i forms an *ear*. Observe that every simple polygon has at least one ear while polygons with holes do not need to have ears. Our proofs in subsequent sections strongly rely on the existence of ears and therefore cannot be generalized to non-simple polygons.

The visibility graph of a polygon has a node for every vertex of the polygon and an edge for every pair of vertices that see each other. We use m to denote the number

of edges in the visibility graph of a given polygon. We define the degree of a vertex of the polygon to be the degree of the corresponding node in the visibility graph.

With $\text{vis}(v_i) = (u_1, \ldots, u_d)$ we denote the sequence of vertices that a vertex $v_i \in V$ of degree d sees, enumerated in counter-clockwise order along the boundary starting at $u_1 = v_{i+1}$ and ending at $u_d = v_{i-1}$. We write $\text{vis}_j(v_i)$, $1 \leq j \leq d$, to denote u_j, $\text{vis}_{-j}(v_i)$ to denote u_{d+1-j} and $\text{vis}_0(v_i)$ to denote v_i itself. For a given sequence $\text{chain}(v_i, v_j)$ we denote by $\text{chain}_v(v_i, v_j)$ the subsequence of $\text{chain}(v_i, v_j)$ containing only the vertices visible to v.

Let $C = (u_0, \ldots, u_{l-1})$ be a cycle of length l in the visibility graph of $\mathcal{P}$. We say C is an *ordered cycle*, if and only if $u_0, \ldots, u_{l-1}$ appear on the boundary of $\mathcal{P}$ in that order (counter-clockwise). As u_i sees u_{i+1} for $0 \leq i \leq l-1$, an ordered cycle C induces a subpolygon of $\mathcal{P}$ with C being the boundary of the subpolygon. Note that C being an ordered cycle implies that the boundary of the induced subpolygon does not self-intersect.

Lemma 1 *Let $\mathcal{P}$ be a simple polygon of size $n \geq 4$. For all $0 \leq i < n$ we have that either the degree of v_i or the degree of v_{i+1} is greater than two.*

Proof As any polygon has a triangulation, two vertices that see each other need to share a third vertex they both see. In our instance two vertices v_i and v_{i+1} thus need to share a third vertex w that they both see. For the sake of contradiction assume there is a vertex v_i such that v_i and v_{i+1} have degree two. It follows that w must be a neighbor of both vertices on the boundary and thus the polygon has to be a triangle. This is a contradiction to the assumption $n \geq 4$. □

Look-Back Robot We now define a basic robot which we use as a basis for more sophisticated models later on. A basic robot r in a polygon $\mathcal{P}$ is modeled to be a moving point which is initially situated at some vertex of $\mathcal{P}$. We allow it to move to the neighboring vertices along the boundary of $\mathcal{P}$, and we write v_r to denote its current location. We say r sees a vertex u if $u \in \text{vis}(v_r)$. While r remains at a vertex v_r, we allow it to sense the degree of v_r and to put the vertices it sees into counter-clockwise order starting with v_{r+1}.[1] Note that the robot has no immediate way of globally identifying the vertices it sees, i.e. knowing their global index with respect to v_r (their index counting from v_r in counter-clockwise order along the boundary). As our intention lies in obtaining a weak model, we do not allow robots to gather sensory information while moving. Similarly, when dealing with multiple robots, we say a robot r sees another robot r_2 if and only if r sees v_{r_2} (or equivalently r_2 sees v_r)—in particular the robot can count the number of robots located at any vertex it sees. We do not impose a limitation on the robots' memory.

A natural extension of the basic robot model lies in allowing a robot to not only move along the boundary, but to any vertex it sees. In general a robot does not know the identity (i.e. the global index[2]) of any vertex but its neighbors. This means that

[1] As long as the vertices visible to the robot remain indistinguishable to it, knowing the counter-clockwise order of the vertices it sees does not provide additional information. The order becomes important as soon as we equip the robot with sensors that make it possible to distinguish between some of the vertices it sees.

[2] We usually set v_0 to be the vertex the robot is initially located at.

any local information it gathered while moving along the boundary may be lost once it decides to move to a distant vertex. In order to get around this problem we further enhance our robot model by adding the look-back capability. To explain what this means, consider a robot at some vertex v_i that drives to a distant visible vertex v_j. Without look-back, the robot afterwards has no way of knowing the identity of v_i in $\mathrm{vis}(v_j)$. Look-back provides the robot the index k such that $v_i = \mathrm{vis}_k(v_j)$. We call such a robot a *look-back robot* or *LB-robot* for short.

3 Reconstruction and Rendezvous in Simple Polygons

As soon as there is one vertex $v^\star$ of the environment that a robot can distinguish from all other vertices, it can easily determine the visibility graph: It can define $v_0 = v^\star$ and identify the global index of any vertex it sees easily by driving there and then along the boundary until it encounters $v^\star$, counting the number of steps along the boundary (cf. [16]). This means that if the robot can perceive any non-symmetrical structure, it can define a unique vertex $v^\star$ and thus reconstruct the visibility graph. In other words, reconstructing the visibility graph (or meeting other identical robots) is an easy task for most polygons. Only highly symmetrical polygons present a challenge. In order to solve the problem in general (i.e. in any polygon), we make use of concepts that were first introduced in the context of distributed networks.

In distributed computing, a network is modeled as a graph whose edges are labeled by so-called *port numbers*, such that edges incident to any node v are assigned distinct labels from the set $\{1, 2, \ldots, d\}$, where d is the degree of v in the graph. Such an edge-labeling is called local orientation or port-numbering (e.g. see [18]). In our model, an LB-robot is capable of putting the vertices it sees into counter-clockwise order, and this order is the same whenever a vertex is visited. Thus the robot has access to a port numbering of the visibility graph G defined as follows. At any node u of G, each incident edge (u, v) is labeled by i such that $\mathrm{vis}_i(u) = v$. The same edge (u, v) is labeled j at the other end-point v, such that $\mathrm{vis}_j(v) = u$. Thus, each edge of G is labeled by two numbers, one at either end-point (as in a bidirectional network with port numbering). Look-back allows the robot to access the port-number of the edge through which it has entered a node v of G. This corresponds to the *Port-to-Port* model of message-passing in distributed networks. In other words, our robot can backtrack any number of moves, or, in the sense of distributed computing, to send messages back and forth between vertices. In distributed networks it has proven beneficial [18] to define the *view* of a vertex as the collection of information a node can gather by sending and receiving messages. We will introduce the same concept in our setting of a robot moving along the visibility graph of a polygon and apply it in order to analyze the capabilities of the robot.

In the following we adapt the definition of the view of a vertex, to suit our robot perspective. Consult Fig. 1 (left) along with the definitions.

Definition 1 Let v be a vertex of a simple polygon $\mathcal{P}$ and d be its degree. The *level-1-label* $l(v) \equiv l^{(1)}(v)$ of v is given by $l(v) := (r_1, \ldots, r_i, \ldots, r_d)$ where r_i is defined such that $\mathrm{vis}_{r_i}(\mathrm{vis}_i(v)) = v$, i.e. r_i is the index of v in $\mathrm{vis}_i(v)$'s list of neighbors.

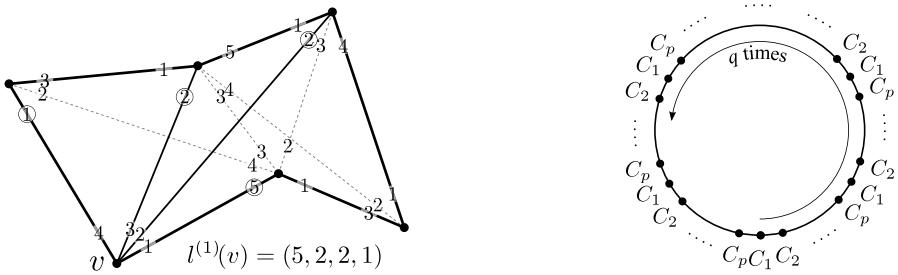

Fig. 1 *Left*: illustration of the definition of the level-1-label of a vertex v. *Right*: illustration of how classes repeat along the boundary of a highly symmetrical polygon

We write $l_j(v)$ to denote r_j and $l_{-j}(v)$ to denote r_{d+1-j} for $0 < j \leq d$. Similarly $l_i(v) = -k$ means $\text{vis}_{-k}(\text{vis}_i(v)) = v$.

Let v be a vertex of a simple polygon $\mathcal{P}$ and d be its degree. Let $\text{vis}(v) = (u_1, \ldots, u_d)$. For $k \in \mathbb{N}, k > 1$, the level-k-label $l^{(k)}(v)$ of v is given by $l^{(k)}(v) := (l^{(k-1)}(v), (l^{(k-1)}(u_1), \ldots, l^{(k-1)}(u_d)))$.

Note that the definition of the level-k-label of a vertex v is equivalent to its view up to depth k as originally defined in [18]. Intuitively it contains the information a backtracking agent can gather by exploring the visibility graph up to a distance of k from v.

In the following we introduce the notion of a level-k-class of a vertex for the equivalence class containing all vertices with the same level-k-labels.

Definition 2 Let v be a vertex of a simple polygon $\mathcal{P}$. The level-k-class $C_v^{(k)}(\mathcal{P})$ of v is the set of vertices of $\mathcal{P}$ that have the same level-k-label as v, including v itself. Formally $C_v^{(k)}(\mathcal{P}) := \{u \in V | l^{(k)}(u) = l^{(k)}(v)\}$. The *class* $C_v(\mathcal{P})$ of v is defined to be the set of vertices that have the same level-j-label for all j: $C_v(\mathcal{P}) := \{u \in V | \forall j \geq 1 : l^{(j)}(u) = l^{(j)}(v)\}$. We drop the argument $\mathcal{P}$ whenever the polygon is clear from the context.

The following lemma summarizes the key properties of classes. In particular it states that a finite exploration depth is sufficient for fully characterizing the class of a vertex.

Lemma 2 *Let v be a vertex of a simple polygon $\mathcal{P}$ of size n. For all $k \geq 1$ we have the following properties*:

1. $C_v^{(k+1)} \subseteq C_v^{(k)}$;
2. $C_v \subseteq C_v^{(k)}$;
3. $C_v^{(n-1)} \subseteq C_v^{(k)}$;
4. $C_v^{(n-1)} = C_v$.

Proof Properties 1 and 2 follow immediately from the definitions of level-k-labels and classes. Property 3 was proven by Norris [15] for general graphs (not only vis-

ibility graphs) using the observation that if there is a vertex v such that for some $k > 1$ we have $C_v^{(k)} \neq C_v^{(k+1)}$, then there is also a vertex u for which $C_u^{(k-1)} \neq C_u^{(k)}$. Property 4 follows from Property 3 and the definition of C_v. □

In [18] it was first shown that all classes have the same size q even in general graphs. The following lemma adds that in the case of polygons, the sequence of classes to which the vertices along the boundary belong is periodical with period $\frac{n}{q}$ (cf. Fig. 1 (right)).

Lemma 3 *Let v_i be a vertex of a simple polygon $\mathcal{P}$ of size n. For all vertices $u \in V$ we have $q := |C_{v_i}| = |C_u|$ and $p := \frac{n}{q}$ is an integer equal to the number of different classes of $\mathcal{P}$. For all integers k we have $C_{v_i} = C_{v_{i+kp}}$.*

Proof For every $v_j \in V$ we have $|C_{v_j}| = |C_{v_{j+1}}|$ as every vertex of C_{v_j} has a vertex of $C_{v_{j+1}}$ as its counter-clockwise neighbor on the boundary and conversely every vertex of $C_{v_{j+1}}$ has a vertex of C_{v_j} as its cw neighbor on the boundary. Since this is true for every vertex v_j, all classes must have the same size $q = |C_{v_i}|$ and thus $p = \frac{n}{q}$ is an integer and there are p classes. As the sequence $(C_{v_l}, C_{v_{l+1}}, \ldots, C_{v_{l-1}})$ is the same for all $v_l \in C_{v_i}$, the vertices of C_{v_i} need to be equally spaced in it. Using that an equal spacing is given by $n/|C_{v_i}| = p$, we get $C_{v_i} = C_{v_i+kp}$ for all integers k. □

We now show that level-1-labels are enough to find the ears of a polygon.

Lemma 4 *Let v_i be a vertex of a simple polygon $\mathcal{P}$ of size $n \geq 3$. We have that v_i is an ear if and only if $l_2(v_{i-1}) = -2$ or equivalently $l_{-2}(v_{i+1}) = 2$.*

Proof If v_i is an ear, we obviously have $\text{vis}_2(v_{i-1}) = v_{i+1}$ and $\text{vis}_{-2}(v_{i+1}) = v_{i-1}$ which implies $l_2(v_{i-1}) = -2$ and $l_{-2}(v_{i+1}) = 2$.

Conversely, $\text{vis}_2(v_{i-1}) = v_{i+1}$ (or $\text{vis}_{-2}(v_{i+1}) = v_{i-1}$) obviously implies that v_i is an ear. As $l_2(v_{i-1}) = -2$ and $l_{-2}(v_{i+1}) = 2$ are equivalent, it remains to show that $l_2(v_{i-1}) = -2$ implies $\text{vis}_2(v_{i-1}) = v_{i+1}$.

For the sake of contradiction assume $w := \text{vis}_2(v_{i-1}) \neq v_{i+1}$. Consider the subpolygon $\mathcal{P}'$ induced by the ordered cycle $\text{chain}(v_{i-1}, w)$. In $\mathcal{P}'$, v_{i-1} and w have degree 2. As $w \neq v_{i+1}$, we have $|\text{chain}(v_{i-1}, w)| \geq 4$ which contradicts Lemma 1. Therefore $\text{vis}_2(v_{i-1}) = v_{i+1}$ and v_i is an ear. □

Lemma 5 *Let v_i be a vertex of a simple polygon $\mathcal{P}$. If v_i is an ear of $\mathcal{P}$, then every vertex in $C_{v_i}^{(2)}$ is an ear.*

Proof By Lemma 4 we know that $l_2(v_{i-1}) = -2$ and $l_{-2}(v_{i+1}) = 2$. Let $v_j \in C_{v_i}^{(2)}$. Because $l^{(2)}(v_i) = l^{(2)}(v_j)$, we have $l(v_{i-1}) = l(v_{j-1})$ and $l(v_{i+1}) = l(v_{j+1})$ from which it follows that $l_2(v_{j-1}) = -2$ and $l_{-2}(v_{j+1}) = 2$. This however implies that v_j is an ear (again using Lemma 4). □

The following lemma allows a robot to 'cut off' ears of the polygon. With cutting off an ear v_i of a polygon $\mathcal{P}$ we mean the operation that removes a vertex v_i yielding

the subpolygon induced by the ordered cycle $v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{n-1}$ in $\mathcal{P}$'s visibility graph. Cutting off a single ear is problematic for a robot as it has no obvious way of deciding which edges of the visibility graph it has to ignore afterwards in order to restrict itself to the remaining subpolygon. An edge might lead to a vertex of the same class as the one the robot cut off, in which case it has no way of distinguishing whether the vertex is still there or not. Cutting off all vertices of one class however is possible as the robot can then simply ignore all edges leading to vertices of the corresponding class altogether.

Lemma 6 *Let v be a vertex of a simple polygon $\mathcal{P}$ of size n with $|C_v| < n$, i.e. $\mathcal{P}$ has more than one class. If v is an ear of $\mathcal{P}$, the subpolygon $\mathcal{P}'$ of $\mathcal{P}$ obtained by cutting off the vertices $C_v(\mathcal{P})$ is well-defined and for all vertices u of $\mathcal{P}'$ we have $C_u(\mathcal{P}) \subseteq C_u(\mathcal{P}')$.*

Proof As v is an ear, Lemma 5 gives us that all vertices in $C_v^{(2)}$ are ears and thus all vertices in $C_v \subseteq C_v^{(2)}$ are (Lemma 2). The subpolygon $\mathcal{P}'$ is thus well-defined as the inducing set of vertices lies on an ordered cycle.

Let u, w be vertices of $\mathcal{P}'$ such that $C_u(\mathcal{P}) = C_w(\mathcal{P})$ and thus for all $k \geq 1$ we have $l^{(k)}(u) = l^{(k)}(w)$ in $\mathcal{P}$. The level-k-label of a vertex v' in $\mathcal{P}'$ can be obtained from its level-k-label in $\mathcal{P}$ by recursively removing all occurrences of labels that belong to a vertex in $C_v(\mathcal{P})$ (i.e. we remove the level-$(k-1)$-labels belonging to vertices in $C_v(\mathcal{P})$ from $l^{(k)}(v')$ and apply the procedure recursively for $k-1$ to the other labels in $l^{(k)}(v')$). As $C_{v'}(\mathcal{P}) = C_{v'}^{(n-1)}(\mathcal{P})$, i.e. a finite depth determines the class of a vertex, the same occurrences are removed for vertices in the same class. Thus in $\mathcal{P}'$, $l^{(k)}(u) = l^{(k)}(w)$ still holds for all $k \geq 1$ and hence $C_u(\mathcal{P}') = C_w(\mathcal{P}')$. □

The following result is the main insight for reconstructing the visibility graph of a polygon and for solving the weak rendezvous problem for multiple robots.

Theorem 1 *For any simple polygon $\mathcal{P}$ there is a vertex v for which $C_v^{(n-1)} = C_v$ forms a clique in the visibility graph of $\mathcal{P}$.*

Proof Using the fact that any simple polygon has an ear, we can select an ear u of our polygon and cut off all vertices in C_u. Lemma 6 gives us that we can do this and we never divide classes in the process. We can repeatedly apply this procedure until we obtain a polygon $\mathcal{P}'$ where every vertex is in the same class. As $\mathcal{P}'$ needs to have at least one ear, it follows from Lemma 5 that all vertices of $\mathcal{P}'$ are ears. This in turn implies that $\mathcal{P}'$ is convex and thus its visibility graph is a complete graph. As we never divide classes in our procedure, we know that there is a vertex v of $\mathcal{P}$ such that every vertex of $C_v(\mathcal{P})$ is still present in $\mathcal{P}'$. Hence, $C_v(\mathcal{P})$ forms a clique in $\mathcal{P}'$ and thus forms a clique in $\mathcal{P}$. □

Lemma 7 *Let $\mathcal{P}$ be a simple polygon of n vertices. Given n and some $k \geq 1$, an LB-robot located at v can determine $l^{(k)}(v)$.*

Proof An LB-robot can compute the level-1-label $l^{(1)}(v)$ of a vertex v by moving to every neighbor of v in turn. It can recursively compute $l^{(k)}(v)$, the level-k-label of v,

by moving to every neighbor of v in turn and computing its level-$(k-1)$-label. In both cases, as the robot can look back, it is capable of returning to v after visiting a neighbor. □

Using Lemma 7 and Theorem 1, we conclude that any number of deterministic LB-robots can position themselves such that they are mutually visible.

Theorem 2 *Let $\mathcal{P}$ be a simple polygon of n vertices. Given n, any number of identical and deterministic LB-robots can weakly meet in $\mathcal{P}$, i.e. they can position themselves such that every robot sees all other robots.*

Proof By Lemma 7 a robot can calculate $l^{(n)}(v)$ for every vertex v along the boundary and thus not only find the classes of all vertices but also the classes of all vertices they see. By Theorem 1 at least one class forms a clique. The robot can easily check which classes form a clique (by comparing the level-n-labels) and drive to a vertex of the class $C_{\min}$ with the lexicographically smallest level-n-label among all these classes. This strategy will choose the same class $C_{\min}$ for every robot independent of its starting location, thus if all robots execute it they will eventually all be located on vertices of $C_{\min}$ seeing each other as $C_{\min}$ forms a clique. □

In the following we again use the fact that an LB-robot can find a class of vertices that forms a clique in the visibility graph, and we show that using this clique as a frame, it can incrementally build up the visibility graph of the polygon.

Theorem 3 *Let $\mathcal{P}$ be a simple polygon of n vertices. Given n, an LB-robot can determine the visibility graph of $\mathcal{P}$.*

Proof By Lemma 7 the robot can determine $l^{(n)}(v)$ for all vertices v along the boundary and thus not only find the classes of all vertices but also the classes of all vertices they see. As in the proof of Theorem 2, this means the robot can identify the class $C_{\min}$ with the lexicographically smallest level-n-label among the classes forming a clique.

Let $u \in C_{\min}$ be a vertex of this class. We argue that the edges of u in the visibility graph of $\mathcal{P}$ are easily identified. Assume that the i-th edge of u (in counter-clockwise order) leads to a vertex w of class C' and let x_i be the number of edges with index $j < i$ that lead to vertices of class $C_{\min}$. If $C' = C_{\min}$, w is easily identified as the (x_i+1)-th vertex of class $C_{\min}$ counting along the boundary starting at u. If $C' \neq C_{\min}$, w is the first vertex of class C' after the x_i-th vertex of class $C_{\min}$ in counter-clockwise order counting from u (by Lemma 3, there is exactly one vertex of class C' between the x_i-th and the (x_i+1)-th vertex of class $C_{\min}$). In the following we show that the robot can identify the edges incident to other vertices ($u \notin C_{\min}$).

Let v_i be a vertex of $\mathcal{P}$. With $d_k(v_i)$ and $d_{-k}(v_i)$ we denote the set of vertices in $\text{chain}_{v_i}(v_i, v_{i+k})$ and $\text{chain}_{v_i}(v_{i-k}, v_i)$, respectively. In terms of this definition, finding the visibility graph of $\mathcal{P}$ is the same as finding $d_{\frac{n}{2}}(v)$ and $d_{-\frac{n}{2}}(v)$ (i.e. finding the global indices of the vertices in $d_{\frac{n}{2}}(v)$ and $d_{-\frac{n}{2}}(v)$) of every vertex v along the boundary of $\mathcal{P}$. In the following we assume: ($\star$) for any two vertices u and w,

$C_u = C_w$ implies $|d_k(u)| = |d_k(w)|$ and $|d_{-k}(u)| = |d_{-k}(w)|$ for all $k \geq 1$. We will prove ($\star$) later, but for now observe that it trivially holds for $k = 1$.

We show inductively how to obtain $d_{\pm\frac{n}{2}}$ for all vertices. The robot knows $d_{\pm 1}$ for every vertex as every vertex sees its neighbors on the boundary. It remains to be shown how to obtain $d_{k+1}(v_i)$ for some vertex v_i assuming d_k is known—this can then be applied to all vertices in order to obtain $d_{\pm(k+1)}$ for all vertices. Let $x := |d_k(v_i)|$ and $\text{vis}(v_i) = (u_1, \ldots, u_d)$, where d is v_i's degree. We have

$$d_{k+1}(v_i) = \begin{cases} d_k(v_i) \cup \{v_{i+k+1}\}, & \text{if } u_{x+1} = v_{i+k+1} \\ d_k(v_i), & \text{otherwise.} \end{cases}$$

Let $v_j := u_{x+1}$ be the first vertex (in counter-clockwise order) visible to v_i and not in $d_k(v_i)$. In order to derive $d_{k+1}(v_i)$, it is now enough for the robot to decide whether $v_j = v_{i+k+1}$ or $v_j \neq v_{i+k+1}$. As the robot knows $C_{v_{i+k+1}}$ and can compute C_{v_j}, this is trivial for $v_{i+k+1} \notin C_{v_j}$. We therefore restrict ourselves to the case $v_{i+k+1} \in C_{v_j}$. We then have $y := |d_{-k}(v_{i+k+1})| = |d_{-k}(v_j)|$, using ($\star$). Assuming $v_j = v_{i+k+1}$ immediately leads to $l_{x+1}(v_i) = -(y+1)$. We want to show that $l_{x+1}(v_i) = -(y+1)$ if and only if $v_j = v_{i+k+1}$. For the sake of contradiction assume $v_j \neq v_{i+k+1}$ and $l_{x+1}(v_i) = -(y+1)$.

Let $a \in C_{\min}$ be the first vertex in $\text{chain}(v_i, v_j)$ lying in $C_{\min}$ and likewise let $b \in C_{\min}$ be the last vertex in $\text{chain}(v_i, v_j)$ lying in $C_{\min}$. Note that a and b are well-defined, as $v_{i+k+1} \in \text{chain}(v_i, v_j)$ and as there is a vertex of $C_{\min}$ in $\text{chain}(v_{i+k+1}, v_j)$ since $C_{v_{i+k+1}} = C_{v_j}$ (Lemma 3). For the same reason we have $b \in \text{chain}(v_{i+k+2}, v_j)$. On the other hand, since the number of vertices of $C_{\min}$ in $\text{chain}(v_{i+1}, v_{i+k+1})$ must be equal to the number of vertices of $C_{\min}$ in $\text{chain}(v_{j-k}, v_j)$, and since $b \notin \text{chain}(v_{i+1}, v_{i+k+1})$, we have $a \notin \text{chain}(v_{j-k}, v_j)$. Hence $a \in \text{chain}(v_{i+1}, v_{j-k-1})$ (cf. Fig. 2).

Consider the case $a \neq b$ (cf. Fig. 2 (left)). We define s to be the last vertex in $\text{chain}(v_{i+1}, a)$ visible to v_i and t to be the first vertex in $\text{chain}(b, v_{j-1})$ visible to v_j. Let $\mathcal{P}'$ be the subpolygon induced by $v_i, \text{chain}(s, a), \text{chain}(b, t), v_j$. This subpolygon is well-defined since a sees b (as $C_{\min}$ forms a clique), and it has at least four vertices. Note that v_i does not see any vertices in $\text{chain}(b, v_{j-1})$, likewise v_j does not see any vertices in $\text{chain}(v_{i+1}, a)$ (recall that $v_j = u_{x+1}$ and $l_{x+1}(v_i) = -(y+1)$). In $\mathcal{P}'$, v_i and v_j are neighbors on the boundary and both have degree 2 which is a contradiction with Lemma 1.

We may thus assume $a = b$ (cf. Fig. 2 (right)). As $a \in \text{chain}(v_{i+1}, v_{j-k-1})$ and $b \in \text{chain}(v_{i+k+2}, v_j)$, this means that $\text{chain}(v_i, v_{i+k+1})$ and $\text{chain}(v_{j-k}, v_j)$ do not overlap. Let now s be the last vertex in $\text{chain}(v_{i+1}, v_{i+k})$ visible to v_i and t be the first vertex in $\text{chain}(v_{j-k}, v_{j-1})$ visible to v_j. We can then define the subpolygon $\mathcal{P}'$ induced by $v_i, \text{chain}(s, t), v_j$ in which again v_i and v_j are neighbors of both degree 2, which is a contradiction to Lemma 1.

We have seen that $l_{x+1}(v_i) = -(y+1)$ is necessary and sufficient for $v_j = v_{i+k+1}$. Both x and y as well as level-1-labels can be derived from $l^{(n)}(v_i)$. This proves that our robot can compute the visibility graph inductively. To complete the proof we still need to show ($\star$). Obviously we have ($\star$) for $k = 1$. By inspection of the inductive method above for $k > 1$, we see that whenever we conclude $v_j = v_{i+k+1}$ for some

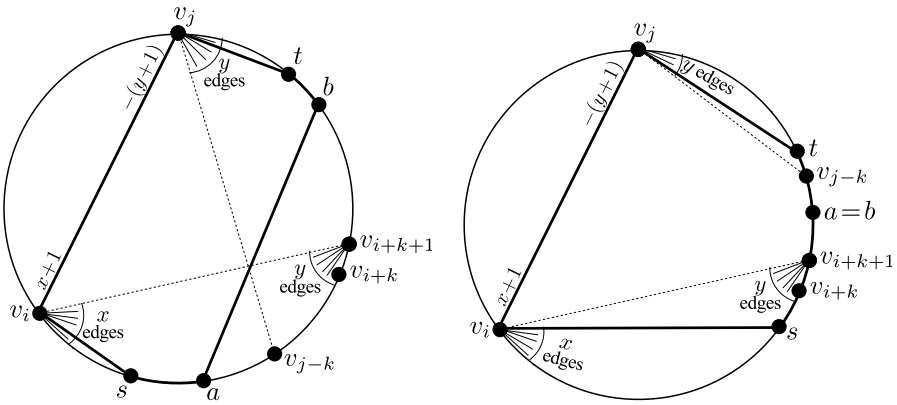

Fig. 2 Illustration of the visibility graph for the cases $a \neq b$ (*left*) and $a = b$ (*right*). The outline of the subpolygon causing a contradiction is highlighted

vertex v_i, the same conclusion will be made for any other vertex $v_l \in C_{v_i}$. In step k of the induction, we only use ($\star$) for $d_{\pm k}$. Hence ($\star$) is maintained throughout. □

The following theorem implies that the results of Theorem 2 and Theorem 3 still hold if only an upper bound $N \geq n$ on the number of vertices is given.

Theorem 4 *Let $\mathcal{P}$ be a simple polygon of n vertices. Given a bound $N \geq n$, an LB-robot can determine the number of vertices n.*

Proof By Lemma 7, the robot can determine the sequence of level-N-labels $l^{(N)}(v_i), l^{(N)}(v_{i+1}), \ldots, l^{(N)}(v_{i+N-1})$, where v_i is its initial location. We have $C^{(N-1)}_{v_j} = C_{v_j}$ (Lemma 2) and thus the robot knows the classes of the vertices $v_i, v_{i+1}, \ldots, v_{i+N-1}$ as well as the classes of the vertices they see. Let $q = |C_{v_i}|$ and $p = \frac{n}{q}$ as in Lemma 3. The robot can easily determine p which is equal to the number of different classes it encounters, since $N \geq n$. Once the robot knows q, it can immediately infer $n = p \cdot q$. By Theorem 1 there is a class $C^\star$ that forms a clique in the visibility graph. Let $M_i := \mathrm{vis}(v_i) \cap C_{v_i}$. As $N \geq n$, there is a vertex of $C^\star$ among $v_i, v_{i+1}, \ldots, v_{i+N-1}$ and hence $q = \max_{i \leq j \leq i+N-1} |M_j| + 1$. As the robot knows the classes of the vertices $v_i, v_{i+1}, \ldots, v_{i+N-1}$ as well as the classes of the vertices they see, it can infer $M_i, M_{i+1}, \ldots, M_{i+N-1}$ and thus q. □

4 Time and Memory Requirements

Until now we have been concerned about the possibility of recognizing the visibility graph of a polygon by a simple look-back robot. Indeed, the main motivation of this paper is to determine what minimum set of capabilities enable the robot to solve the polygon reconstruction and weak rendezvous problems. In this section, we consider the time and space complexity of our algorithms. The cost of an algorithm for a robot is measured in terms of the number of moves made by the robot, where one move

corresponds to the traversal of an edge in the visibility graph.[3] For the algorithm we described in the last section, the number of moves made by the robot is exponential in n. In the following we will show how to adapt the algorithm such that the reconstruction of visibility graphs can be achieved after a polynomial number of moves. Note that map construction of arbitrary unlabeled graphs is not always possible, and even for those cases where it is possible no fast algorithms are known.

Efficient Reconstruction of Visibility Graphs We have shown how an LB-robot can reconstruct the visibility graph of a simple polygon, when the number of vertices n (or at least an upper bound $N \geq n$) is known. The algorithm described in Sect. 3 constructs the visibility graph by computing the level-n-label of each vertex in a recursive manner. Notice that the level-n-label of a vertex has an exponential size in terms of n. However, it is not necessary for the robot to explicitly construct and remember the level-n-labels of the vertices. If the robot can determine the level-n-class for each vertex v along the boundary of the polygon (and for each vertex visible to v), it has sufficient information to construct the visibility graph.

We now present an iterative procedure to compute the level-k-class of each vertex v, for $k = 1$ to n. For $k = 1$, the robot can compute the level-1-label of each vertex in the same manner as before and thus determine the level-1-classes. This computation requires the robot to walk along the boundary of the polygon, and at each vertex v to visit each visible neighbor of v and return back. Let us call this the *basic traversal*. During the basic traversal, the robot makes $(2m - n)$ moves, since each edge is traversed twice, except for the boundary edges which need to be traversed only once. (If n is not known then a basic traversal requires no more than $2\lceil N/n \rceil m$ edge traversals.) Suppose that for some $k \geq 1$ the robot has already determined the level-x-class of each vertex v_i, for all $x \leq k$. We show that using this information, the robot can efficiently compute the level-$(k+1)$-class of any vertex v, since the robot can distinguish between vertices belonging to distinct level-k-classes, as explained below.

Recall that each edge (u, v) of the visibility graph is labeled with two port numbers, one at each end-point of the edge. When a robot traverses an edge (u, v) from u to v, it encounters the labels i, j (in this order) where $\mathrm{vis}_i(u) = v$ and $\mathrm{vis}_j(v) = u$. For any path $(u_0, u_1, \ldots, u_t)$ in the visibility graph G, there exists a corresponding label-sequence, i.e. a sequence of ordered pairs of port numbers, for the edges (u_0, u_1), (u_1, u_2) and so on.

Let $C_1^k, C_2^k, \ldots, C_{n_k}^k$ be the distinct level-k-classes (where n_k is the number of distinct level-k-classes). We define below the concept of *distinguishing paths* for any pair of distinct classes C_i^k, C_j^k.

Definition 3 For any two vertices x and y that belong to distinct level-k classes C_i^k and C_j^k, we define the *distinguishing-path*, $DP^k[x/y]$ to be a sequence of $l \leq k$ ordered pairs of numbers that corresponds to the port-numbers on a path of length l starting from vertex $x \in C_i^k$, such that no path starting from vertex $y \in C_j^k$ has the same sequence of port numbers.

[3]For simplicity, let us assume that the time taken by the robot to move to any visible vertex is bounded by a constant. Then the time complexity of an algorithm is proportional to the number of moves.

It is easy to see that for any two vertices x and y that belong to distinct level-k-classes C_i^k and C_j^k respectively, there always exists at least one distinguishing-path which can be either $DP^k[x/y]$ or $DP^k[y/x]$. (If neither of them exists then the set of paths of length k starting from x is identical to those starting from y, and this would imply that x belongs to the same level-k-class as y—a contradiction.) If there is more than one distinguishing path for the pair of vertices x and y, then it is possible to choose a unique distinguishing path, e.g. the minimum one in the lexicographical ordering of the sequences. This unique distinguishing path which differentiates the vertices of the class C_i^k from the vertices of the class C_j^k is denoted by $DP^k(i, j)$.

For the n_k classes at level k, there are $\binom{n_k}{2}$ such distinguishing paths (one for each pair of distinct classes). An LB-robot can construct the $\binom{n_k}{2}$ distinguishing paths using the same iterative procedure as for constructing the classes. We first show how the distinguishing paths can be used for determining the class of a vertex.

Lemma 8 *An LB-robot that knows the distinguishing paths at level k can determine the level-k-class C_t^k of any arbitrary vertex u_t by traversing at most n_k paths, each of length at most k, starting from u_t.*

Proof An LB-robot placed at vertex u_t can determine the level-k-class of this vertex by attempting to traverse the distinguishing paths for level k, starting from u_t. Whenever the robot tries a distinguishing path $DP^k(i, j)$ it can eliminate one of the two classes C_i^k or C_j^k. Thus after traversing $n_k - 1$ distinguishing paths, the robot knows exactly which class the vertex u_t belongs to. □

The distinguishing paths can be computed inductively. The level-1 distinguishing paths correspond to single edges and the LB-robot can construct these by performing a basic traversal. The distinguishing paths for level k, $k > 1$, are obtained using the distinguishing paths for level $(k - 1)$.

Lemma 9 *An LB-robot can construct the distinguishing path $DP^k(i, j)$ for any two distinct level-k-classes C_i^k and C_j^k, without making any additional moves if it already knows*

(i) *the distinguishing paths for every pair of classes at level $(k - 1)$, and*
(ii) *for each vertex v along the boundary of the polygon, level-1-label and level-k-class of v and the level-$(k - 1)$-class of each neighbor of v.*

Proof Suppose the robot knows all distinguishing paths at level $(k - 1)$. In order to compute the distinguishing path $DP^k(i, j)$, consider two vertices u and v belonging to the distinct level-k-classes C_i^k and C_j^k and inspect their level-1-labels. If there is a difference in the level-1-labels, then we immediately obtain a distinguishing path of length one. Otherwise, there exists an integer $q > 0$ such that the q-th neighbor of u and the q-th neighbor of v belong to distinct level-$(k - 1)$-classes. Thus, using the level-$(k - 1)$ distinguishing path for these two classes and prefixing it with the port-numbers of the q-th incident edge of u, one can obtain the required distinguishing path $DP^k(i, j)$. □

Theorem 5 *Given a bound $N \geq n$ on the size of the polygon, an LB-robot can compute the class $C_v = C_v^n$ for each vertex v along the boundary of the polygon and the class C_{u_i} for each neighbor u_i of v, using a total of $O(mNn^2)$ moves.*

Proof We already know that the robot can construct the level-1-classes and the level-1 distinguishing paths in $O(mN/n)$ moves. For $k > 1$, the knowledge of level-$(k-1)$-classes and the corresponding distinguishing paths allows the LB-robot to construct the level-k-class of each vertex v of the polygon (see Lemma 8). The robot performs a basic traversal and whenever it reaches a neighbor u_i of a vertex v, it determines the level-k-class of u_i by traversing at most n paths each of length k starting from u_i. Since the robot repeats this process for each vertex of the polygon (while traversing along the boundary), it makes at most $N + \sum_{i=1}^{N} \deg(v_i) \cdot n \cdot k = O(m \cdot N \cdot k)$ moves for computing the level-k-classes of all the vertices, where $\deg(v_i)$ is the degree of v_i in the visibility graph. Thus the number of moves made in total for computing the level-k-labels for $k = 1$ to n, using the above procedure, is in $O(m \cdot N \cdot \sum_{k=1}^{n} k)$ which is in $O(mNn^2)$. Note that the robot does not need to continue until level $k = N$; once the robot reaches level $k = n$, it can detect the class $C^\star$ that forms a clique in the visibility graph and thus it can infer the value of n. □

Due to the observations made in the previous section, the above result immediately implies the following:

Corollary 1 *Given $N \geq n$, the weak rendezvous problem can be solved after at most $O(mNn^2)$ moves by each LB-robot.*

Corollary 2 *Given $N \geq n$, an LB-robot can construct the visibility graph of a simple polygon $\mathcal{P}$ of n vertices, in $O(mNn^2)$ time steps.*

Memory Required for Constructing Visibility Graphs The space complexity of an algorithm for a robot is measured in terms of the amount of data the robot has to carry when moving from one vertex to another. Let us now consider how much memory a robot needs in order to construct the visibility graph of any polygon and to solve weak rendezvous. Recall that the level-k-labels of the vertices have lengths exponential in k. Since the robot does not explicitly remember the level-k-labels of the vertices, the memory requirement for our algorithm is still polynomial in n (and N). At any level k of the computation, the robot needs to remember the distinguishing paths for that level. There are at most n^2 such paths and each can be stored using $O(n \log n)$ bits of memory. Further, for each vertex v visited during a basic traversal, the robot needs to remember a sequence of indices $j_0, j_1, \ldots, j_d$ such that v belongs to class $C_{j_0}^k$, and the ith neighbor of v belongs to class $C_{j_i}^k$. This requires an additional $O(Nd \log n)$ bits of memory where d is the maximum degree of a vertex in the visibility graph. Hence, we have the following result.

Theorem 6 *LB-robots each knowing N and having $O((n^3 + Nd) \log n)$ bits of memory can construct the visibility graph and solve the weak rendezvous problem in any simple polygon $\mathcal{P}$ of at most $n \leq N$ vertices.*

5 Summary and Future Work

In this work, we have introduced the notion of an LB-robot; a model that allows the robot in a polygon to move to any vertex it sees and to identify which vertex in its current view it came from in its last move. We have shown that, given a bound on the number of vertices, LB-robots can solve the weak rendezvous problem and are capable of reconstructing the visibility graph of the polygon in polynomial time. To show this, we adopted the concept of views from distributed computing (cf. [18]) into our notion of the level-k-label of a vertex with the corresponding level-k-class containing all vertices that have the same level-k-label. Our central result is the fact that at least one class forms a clique in the polygon's visibility graph. From this result we concluded that computing the level-n-labels of all vertices is essentially enough for multiple robots to weakly meet and, with some additional effort, also to reconstruct the visibility graph of the polygon. Our solution improves on the previous result [4] where LB-robots additionally were allowed to measure angles very roughly (in fact only to distinguish between angles smaller and larger than π). The results in the present paper show that this additional capability is not necessary. A recent complimentary result [10] shows that the ability of measuring angles accurately is in itself sufficient to solve the problem (without the look-back capability considered in this paper). This still leaves open the question of whether it is possible to find a "weaker" robot model that still allows the reconstruction of the visibility graph.

Acknowledgements We wish to thank a reviewer for comments that helped to improve the quality of the paper.

References

1. Alpern, S., Gal, S.: The Theory of Search Games and Rendezvous. Kluwer Academic, Dordrecht (2003)
2. Ando, H., Oasa, Y., Suzuki, I., Yamashita, M.: Distributed memoryless point convergence algorithm for mobile robots with limited visibility. IEEE Trans. Robot. Autom. **15**(5), 818–828 (1999)
3. Angluin, D.: Local and global properties in networks of processors. In: Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing, pp. 82–93 (1980)
4. Bilò, D., Disser, Y., Mihalák, M., Suri, S., Vicari, E., Widmayer, P.: Reconstructing visibility graphs with simple robots. In: Proceedings of the 16th International Colloquium on Structural Information and Communication Complexity, pp. 87–99 (2009)
5. Boldi, P., Vigna, S.: An effective characterization of computability in anonymous networks. In: Proceedings of the 15th International Conference on Distributed Computing, pp. 33–47 (2001)
6. Brunner, J., Mihalák, M., Suri, S., Vicari, E., Widmayer, P.: Simple robots in polygonal environments: a hierarchy. In: Proceedings of the Fourth International Workshop on Algorithmic Aspects of Wireless Sensor Networks, pp. 111–124 (2008)
7. Chalopin, J., Das, S., Disser, Y., Mihalák, M., Widmayer, P.: How simple robots benefit from looking back. In: Proceedings of the 7th International Conference on Algorithms and Complexity, pp. 229–239 (2010)
8. Chalopin, J., Godard, E., Métivier, Y., Ossamy, R.: Mobile agent algorithms versus message passing algorithms. In: Proceedings of the 10th International Conference on the Principles of Distributed Systems, pp. 187–201 (2006)
9. Cohen, R., Peleg, D.: Convergence of autonomous mobile robots with inaccurate sensors and movements. SIAM J. Comput. **38**(1), 276–302 (2008)
10. Disser, Y., Mihalák, M., Widmayer, P.: Reconstructing a simple polygon from its angles. In: Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory, pp. 13–24 (2010)

11. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of asynchronous robots with limited visibility. Theor. Comput. Sci. **337**(1–3), 147–168 (2005)
12. Ganguli, A., Cortés, J., Bullo, F.: Distributed deployment of asynchronous guards in art galleries. In: Proceedings of the 2006 American Control Conference, pp. 1416–1421 (2006)
13. Ghosh, S.K.: Visibility Algorithms in the Plane. Cambridge University Press, Cambridge (2007)
14. Kranakis, E., Krizanc, D., Rajsbaum, S.: Mobile agent rendezvous: a survey. In: Proceedings of the 13th International Colloquium Structural Information and Communication Complexity, pp. 1–9 (2006)
15. Norris, N.: Universal covers of graphs: isomorphism to depth $n-1$ implies isomorphism to all depths. Discrete Appl. Math. **56**(1), 61–74 (1995)
16. Suri, S., Vicari, E., Widmayer, P.: Simple robots with minimal sensing: from local visibility to global geometry. Int. J. Robot. Res. **27**(9), 1055–1067 (2008)
17. Suzuki, I., Yamashita, M.: Distributed anonymous mobile robots: formation of geometric patterns. SIAM J. Comput. **28**(4), 1347–1363 (1999)
18. Yamashita, M., Kameda, T.: Computing on anonymous networks: Part I—characterizing the solvable cases. IEEE Trans. Parallel Distrib. Syst. **7**(1), 69–89 (1996)
19. Yershova, A., Tovar, B., Ghrist, R., LaValle, S.M.: Bitbots: Simple robots solving complex tasks. In: Proceedings of the 20th National Conference on Artificial Intelligence, vol. 3, pp. 1336–1341 (2005)